

Engenharia de Software II

Profº Ramon Trigo

Introdução ao UML

UML (Unified Modeling Language – linguagem de modelagem unificada) é “uma linguagem-padrão para descrever/documentar projeto de software. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software-intensivo”.

Em outras palavras, assim como os arquitetos criam plantas e projetos para ser usados por uma empresa de construção, os arquitetos de software criam diagramas UML para ajudar os desenvolvedores de software a construir o software.

Se você entender o vocabulário da UML (os elementos visuais do diagrama e seus significados), pode facilmente entender e especificar um sistema e explicar o projeto daquele sistema para outros interessados.

O que é UML



O grande problema do desenvolvimento de novos sistemas utilizando a orientação a objetos nas fases de análise de requisitos, análise de sistemas e design é que não existe uma notação padronizada e realmente eficaz que abranja qualquer tipo de aplicação que se deseje.

Cada simbologia existente possui seus próprios conceitos, gráficos e terminologias, resultando numa grande confusão, especialmente para aqueles que querem utilizar a orientação a objetos não só sabendo para que lado aponta a seta de um relacionamento, mas sabendo criar modelos de qualidade para ajudá-los a construir e manter sistemas cada vez mais eficazes.

O que é UML



Quando a “Unified Modeling Language” (UML) foi lançada, muitos desenvolvedores da área da orientação a objetos ficaram entusiasmados já que essa padronização proposta pela UML era o tipo de força que eles sempre esperaram.

A UML é muito mais que a padronização de uma notação. É também o desenvolvimento de novos conceitos não normalmente usados.

Por isso e muitas outras razões, o bom entendimento da UML não é apenas aprender a simbologia e o seu significado, mas também significa aprender a modelar orientado a objetos no estado da arte



UML foi desenvolvida por Grady Booch, James Rumbaugh, e Ivar Jacobson que são conhecidos como “os três amigos”.



Por que Modelar um Software

Qual a real necessidade de se projetar um casa? Um pedreiro experiente não e capaz de construí-la sem um projeto?

Para se construir um edifício é necessário, em primeiro lugar, desenvolver um projeto muito bem-elaborado, cujos cálculos têm de estar corretos e precisos.

Necessário fornecer uma estimativa de custos, determinar em quanto tempo a construção estará concluída, especificar a quantidade de material a ser adquirida para a construção escolher o local onde o prédio será erguido.

Grandes projetos não podem ser modelados de cabeça.

Por que Modelar um Software

Um sistema de informação precisa ter uma documentação extremamente detalhada, precisa e atualizada para que possa ser mantido com facilidade, rapidez e correção, sem produzir novos erros ao corrigir os antigos.

Modelar um sistema é uma forma bastante eficiente de documentá-lo, mas a modelagem não serve apenas para isso. A documentação é apenas uma das vantagens fornecidas pela modelagem

Modelo de Software

Um modelo de software captura uma visão de um sistema físico é uma abstração do sistema com um certo proposito, como descrever aspectos estruturais ou comportamentais do software.

Assim um modelo descreve completamente aqueles aspectos do sistema físico que são relevantes ao proposito do modelo, no nível apropriado de detalhe.

Dessa forma, um modelo de casos de uso fornecerá uma visão dos requisitos necessários ao sistema, identificando as funcionalidades do software e os atores que poderão utilizá-las, não se preocupando em detalhe nada além disso.

Modelo de Software

Diagramas UML

A UML divide os diagramas em duas categorias: estruturais e comportamentais.

Os diagramas estruturais podem ser utilizados para documentar a organização física, como um objeto se relaciona com outro.

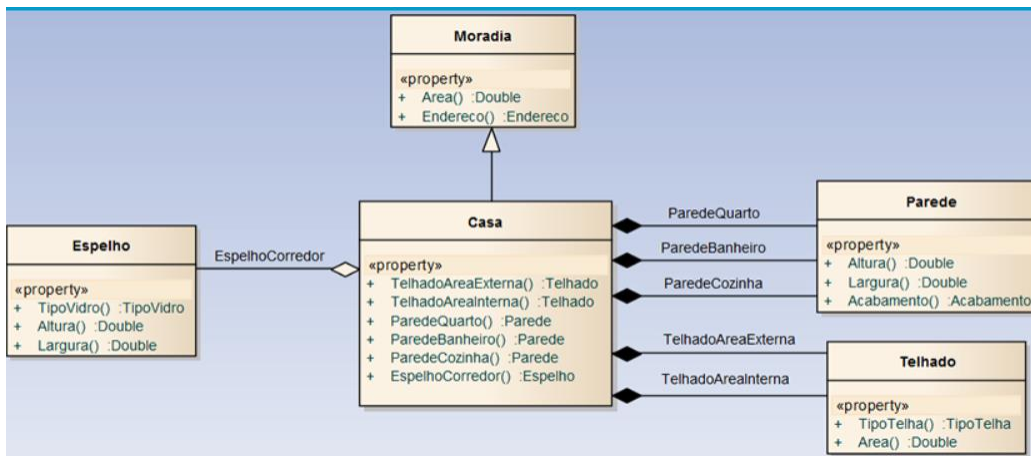
Os diagramas comportamentais tem foco no comportamento dos elementos de um sistema.

Por exemplo, para documentar requisitos, operações e alterações internas de estado para elementos.

Estrutural

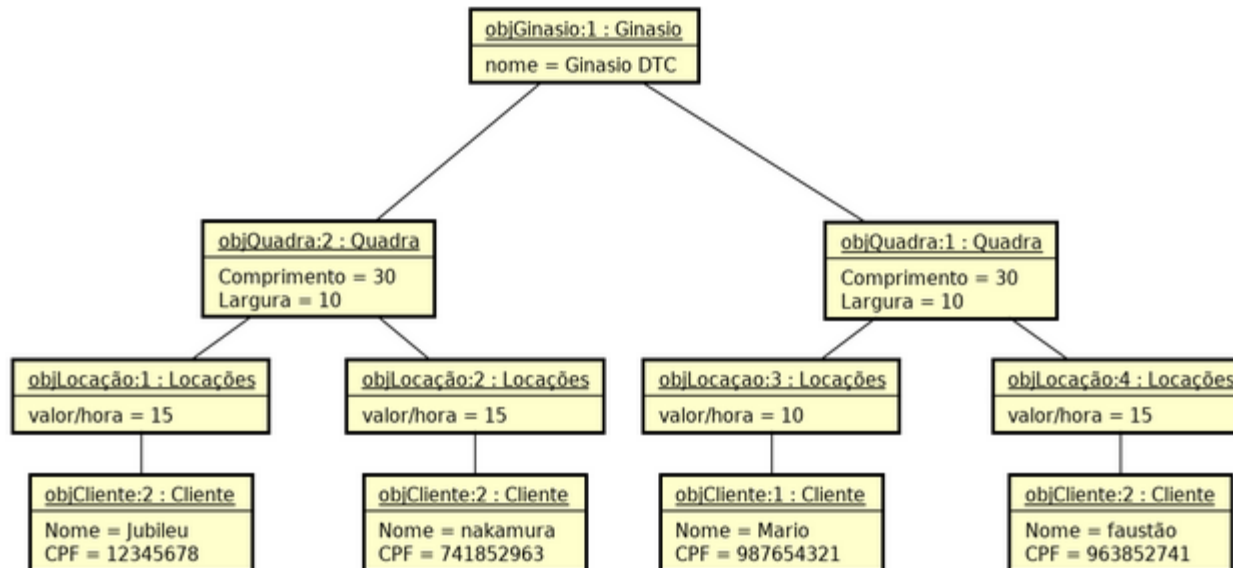
Diagrama de Classe: utilizam classes e interfaces para documentar detalhes sobre as entidades que formam o sistema e as relações estáticas entre elas.

Os diagramas de classe estão entre os mais utilizados e podem variar em detalhes que podem ser depurados e aptos para gerar código fonte, a esboços rápidos em quadros brancos.



Estrutural

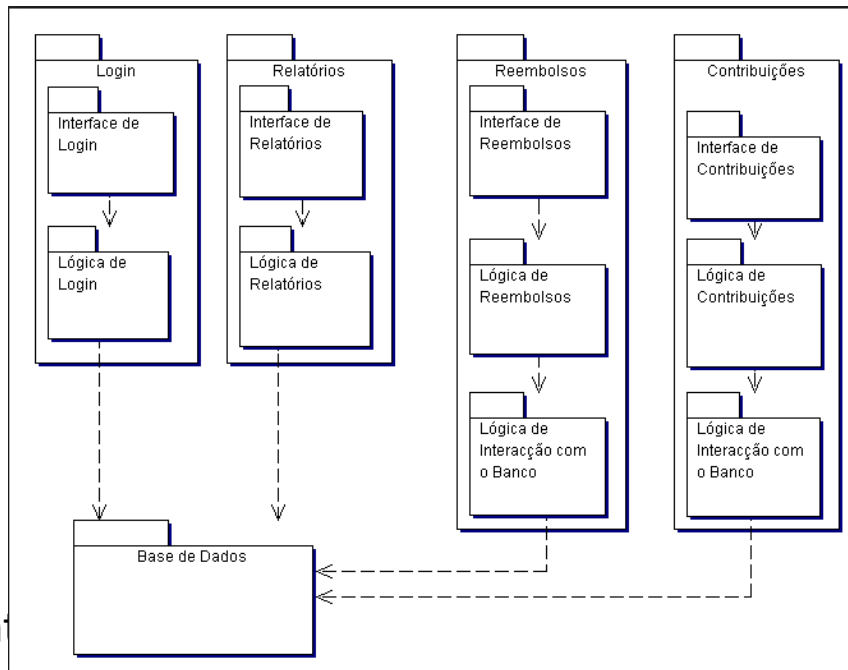
Diagramas de Objeto: utilizam a mesma sintaxe que os diagramas de classes e mostram como as etapas reais de classe são relacionadas num instante específico. Você pode usar os diagramas de objetos para mostrar as relações dentro do sistema em momento específico.



Fonte: <https://medium.com/documenta%C3%A7ao-uml/introdu%C3%A7%C3%A3o-ao-diagrama-de-objetos-902795d485f8>

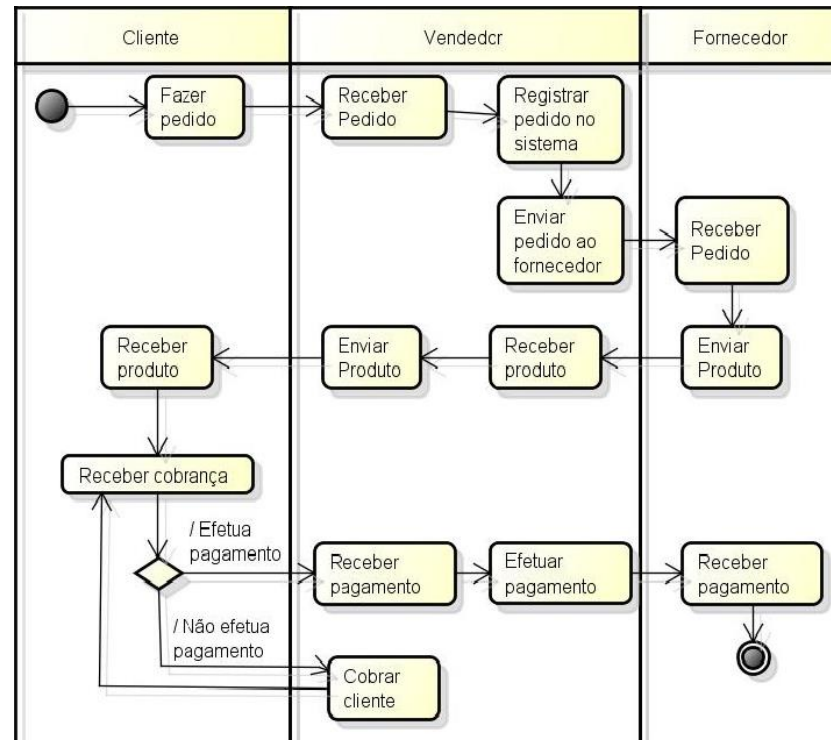
Estrutural

Diagrama de pacote: são realmente tipos especiais de diagramas de classe. Eles utilizam a mesma notação, mas o foco está em como as classes e interfaces são agrupadas.



Comportamental

Diagramas de atividade Fluxos de trabalho de negócios ou operacionais representados graficamente para exibir a atividade de qualquer parte ou componente do sistema. Diagramas de atividade são usados como alternativa aos diagramas de máquina de estados.

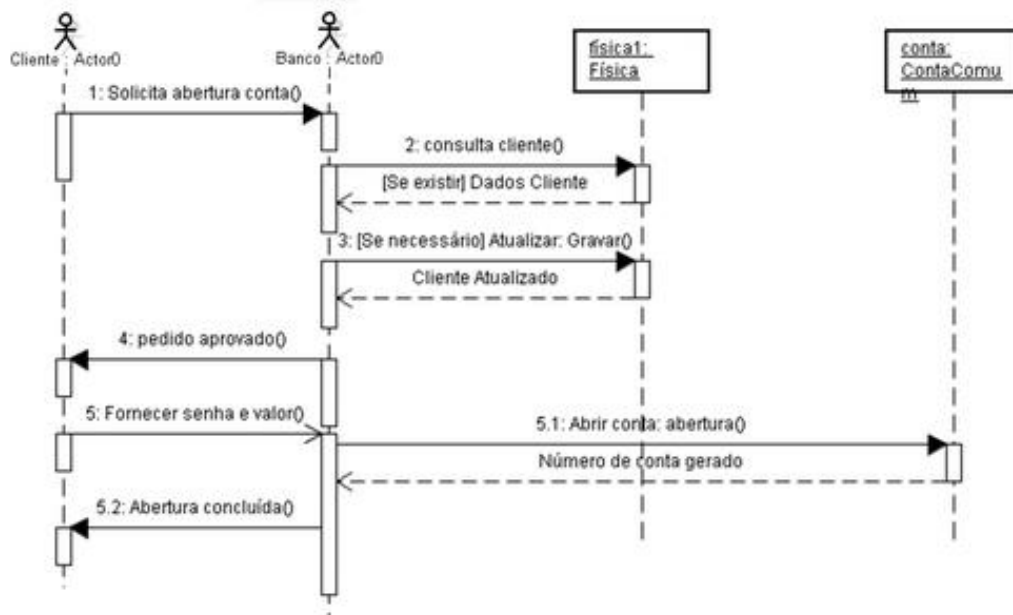


Fonte:

<https://www.researchgate.net/profile/Ilderto-Rodello/publication/280319043/figure/fig2/AS:366637279662081@1464424547133/Figura-2-Diagrama-de-Atividades-modelado-pelo-Grupo-2-antes-da-utilizacao-do-sistema.png>

Comportamental

Diagramas de sequência: tipo de diagrama de interação que enfatiza o tipo e a ordem das mensagens passadas entre os elementos durante a execução. Os diagramas de sequência são o tipo mais comum de diagramas de interação, além de ser muito intuitivos para os novos usuários de UML.



Fonte: <https://medium.com/documenta%C3%A7ao-uml/introdu%C3%A7%C3%A3o-ao-diagrama-de-sequ%C3%Aancia-1ea5e9563594>

Comportamental

Diagramas de caso de uso: documentam os requisitos funcionais para um sistema. Eles fornecem uma visão independente de implementação e do papel que o sistema deve fazer, permitindo ao modelador manter o foco nas necessidades do usuário, ao invés de tê-los nos detalhes de realização.

Por que tantos Diagramas?

O objetivo disso é fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos, procurando-se assim, atingir a completitude da modelagem, permitindo que cada diagrama complemente os outros.

É como se o sistema fosse modelado em camadas, sendo que alguns diagramas enfocam o sistema de forma mais geral, apresentando uma visão externa do sistema, como é o objetivo do Diagrama de Casos de Uso, enquanto quanto outros oferecem um visão de uma camada mais profunda do software.

A utilização de diversos diagramas permite que falhas sejam descobertas, diminuindo a possibilidade da ocorrência de erros futuros.

EXEMPLOS

A construção de um edifício, percebemos que ao se projetar uma construção, esta não tem apenas uma planta, mas diversas, enfocando o projeto de construção do prédio sob diferentes formas, algumas referentes ao layout dos andares, outras apresentando a planta hidráulica e outras ainda abordando a planta elétrica. Isso torna o edifícios completo.

Diagrama de Casos de Uso



O diagrama de casos de uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas.

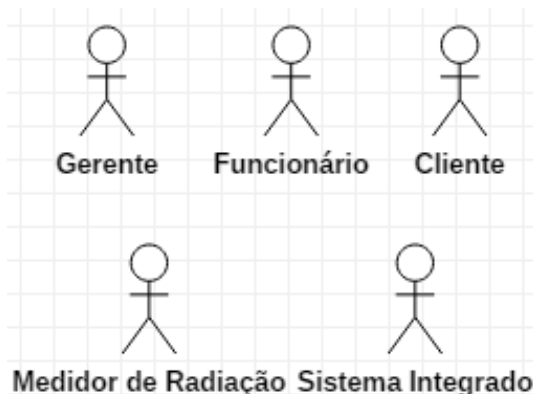
Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar.

Procura identificar os atores (usuários, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, bem como os serviços ou seja as funcionalidades que o sistema disponibilizará aos atores, conhecidas nesse diagrama como casos de uso.

Atores

O diagrama de casos de uso concentra-se em dois itens principais: atores e casos de uso. Os atores representam os papéis desempenhados pelos diversos usuários que poderão utilizar, de alguma maneira, os serviços e funções do sistema.

Os atores pode ser qualquer elemento externo que interaja com o software. Os atores são representados por símbolos de “bonecos magros”, contendo uma breve descrição logo abaixo de seu símbolo que identifica o papel que o ator em questão assume dentro do diagrama



Caso de Uso

Os casos de uso são utilizados para capturar os requisitos do sistema, ou seja referem-se aos serviços, tarefas ou funcionalidades identificados como necessários ao software e que podem ser utilizados de alguma maneira pelos atores que interagem com o sistema, sendo usados para expressar e documentar os comportamentos pretendidos para as funções deste.

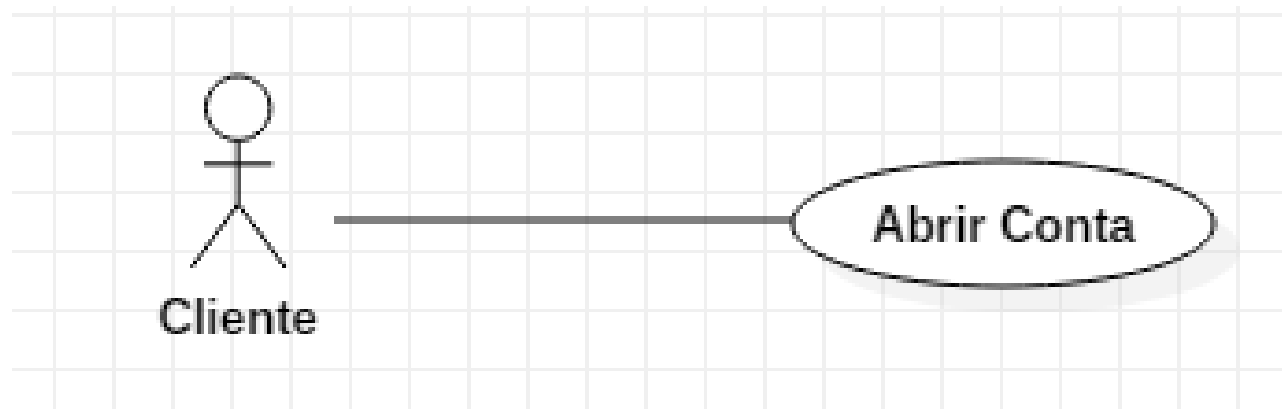
Podem ser classificados em casos de uso primários ou secundários. Um caso de uso é considerado primário quando se refere a um processo importante, que enfoca um dos requisitos funcionais do software.

Caso de uso secundário se refere-se a um processo periférico



Abrir Conta

Exemplo Associação



Generalização/Especialização

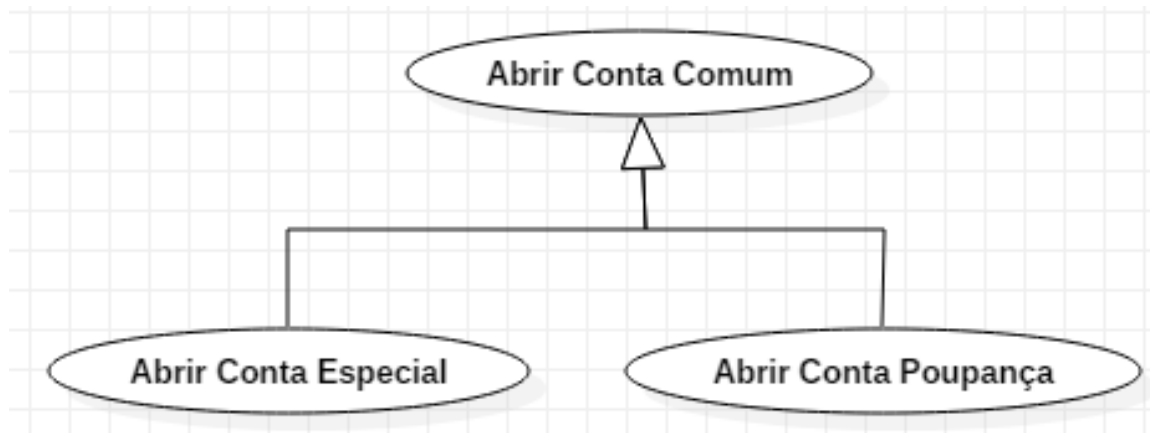
O relacionamento de generalização é uma forma de associação entre os casos de uso na qual existem dois ou mais casos de uso com características semelhantes, apresentando pequenas diferenças entre si.

Além disso, os casos de uso especializados herdam também quaisquer possíveis associações de inclusão ou extensão que o caso de uso geral venha a ter, bem como quaisquer associações com os atores que utilizem o caso de uso geral.

Associação de generalização/especialização é representada por uma linha com uma seta mais grossa, que indica qual o caso de uso geral (para qual a seta aponta).

Generalização/Especialização

Exemplo

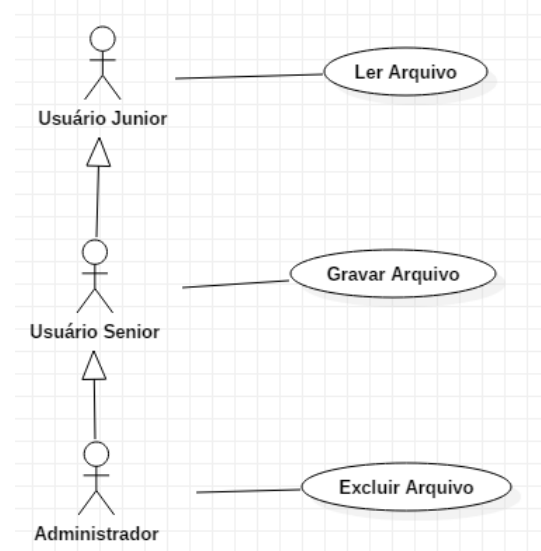
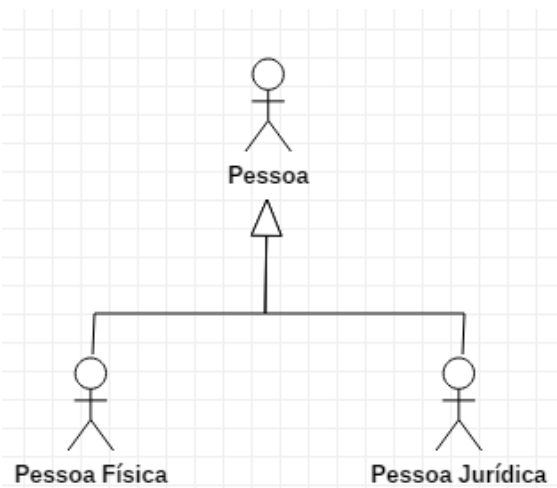


Generalização/Especialização

No exemplo, há três opções de abertura de conta muito semelhante entre si: abertura de conta comum, abertura de conta especial e abertura de conta-poupança, cada uma representada por um caso de uso diferente.

Os processos de abertura de conta especial e de conta poupança são muito semelhantes ao de abertura de conta comum, mas tem algumas características próprias, o que justifica colocá-las como especialização do caso de uso Abrir Conta Comum;

Outros Exemplos



Include

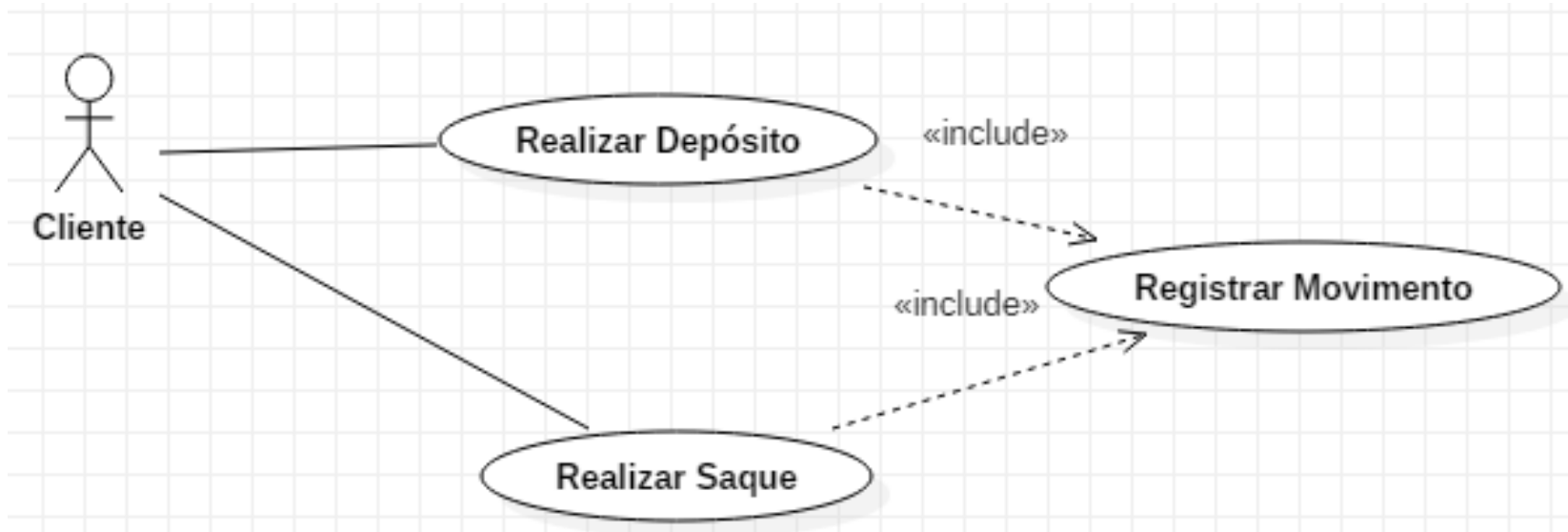
A associação de inclusão costuma ser utilizada quando existe um cenário, situação ou rotina comum a mais de um caso de uso.

Os relacionamentos de inclusão indicam uma obrigatoriedade, ou seja, quando um determinado caso tem um relacionamento de inclusão com outro, a execução do primeiro obriga também a execução do segundo.

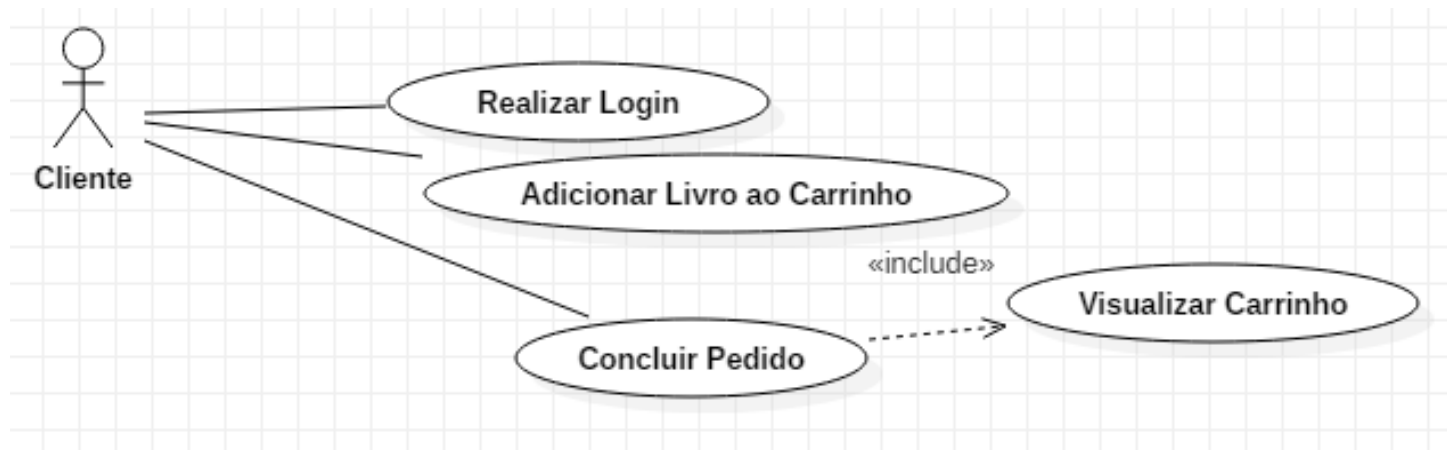
Uma associação de inclusão é representada por uma linha tracejada contendo uma seta em uma de suas extremidades, a qual aponta para o caso de uso incluído no caso de uso posicionado na outra extremidade de linha.

As associações de inclusão costumam apresentar também o texto “include”, entre os sinais < >.

Exemplo



Exemplo



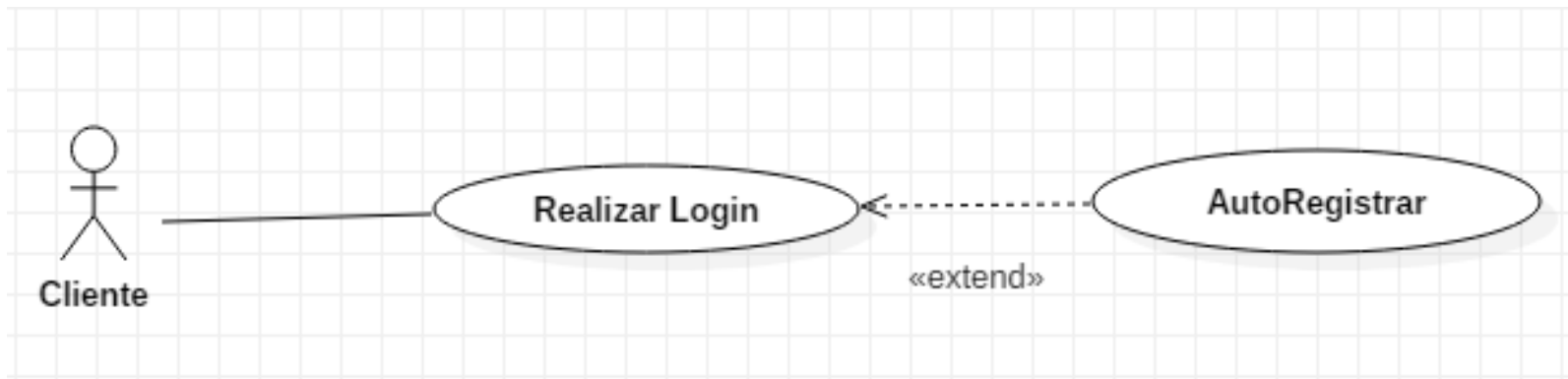
Extends

Associações de extensão são utilizadas para descrever cenários opcionais de um caso de uso.

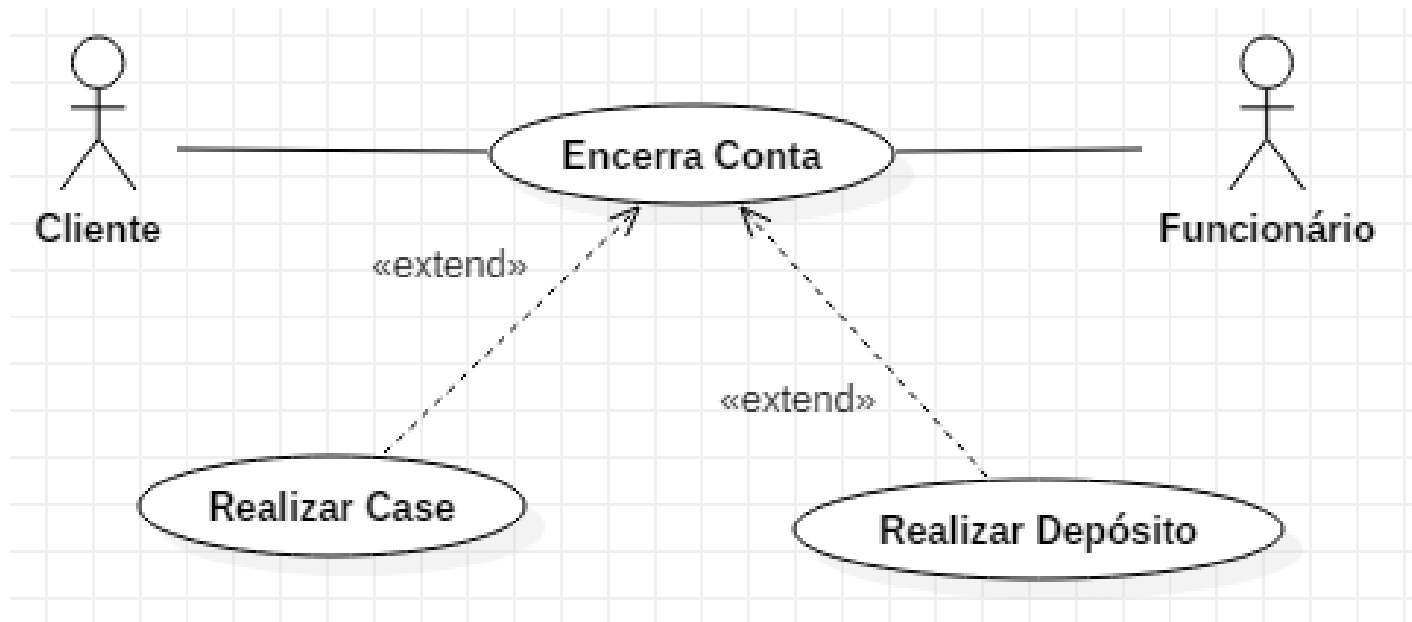
Os casos de uso estendidos descrevem cenários que apenas ocorrerão em uma situação específica se determinada condição for satisfeita.

Assim, as associações de extensão indicam a necessidade de um teste para determinar se é necessário executar também o caso de uso estendido ou não.

Exemplo



Exemplo



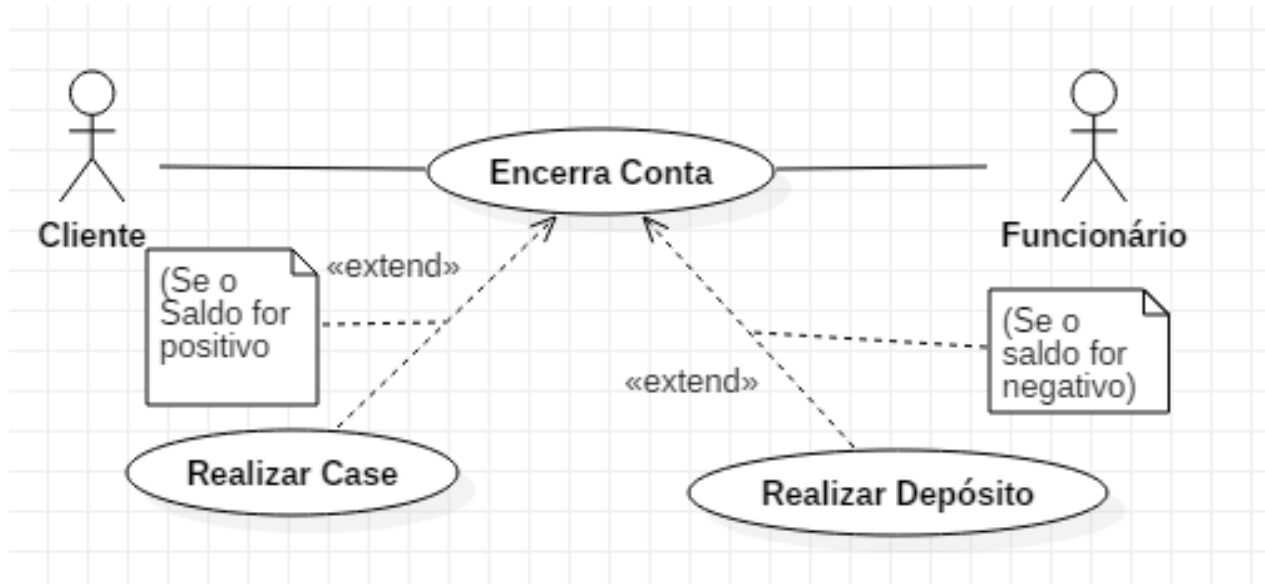
Restrições em Associações de Extensão



Restrições são compostas por um texto entre chaves e utilizadas para definir validações, consistências, condições etc, que devem ser aplicadas a um determinado componente ou situação.

Em se tratando de extensões, às vezes nem sempre fica claro qual é a condição para que um caso de uso estendido seja executado, assim pode-se acrescentar uma restrição à associação de extensão por meio de uma nota explicativa, determinando a condição para que o caso de uso seja executado.

Exemplo



Multiplicidade no Diagrama de Casos de Uso



A multiplicidade em uma associação entre um ator em um caso de uso basicamente especifica o número de vezes que um ator pode utilizar determinado caso de uso



Multiplicidade no Diagrama de Casos de Uso



No exemplo, percebemos que um ator Sócio utiliza o caso de uso Cadastrar Sócio somente uma vez, enquanto o ator Funcionário pode utilizá-los muitas vezes



Exercício

Crie um diagrama de casos de usos

O estudo de caso é sobre uma faculdade que precisa de uma aplicação para controlar alguns processos acadêmicos, como inscrições em disciplinas, lançamentos de notas, alocação de recursos para turmas etc.

Após o levantamento de requisitos inicial deste sistema, os analistas chegaram a seguinte lista de requisitos funcionais:

Exercício

- ✓ O sistema deve permitir que os alunos visualizem as notas obtidas por semestre letivo.
- ✓ O sistema deve permitir o lançamento das notas das disciplinas lecionadas em um semestre letivo e controlar os prazos e atrasos nestes lançamentos.
- ✓ O sistema deve manter informações cadastrais sobre disciplinas no currículo escolar.
- ✓ O sistema deve permitir a abertura de turmas para uma disciplina, assim como a definição de salas e laboratórios a serem utilizadas e dos horários e dias da semana em que haverá aula de tal turma.
- ✓ O sistema deve permitir que os alunos realizem a inscrição em disciplinas de um semestre letivo.
- ✓ O sistema deve permitir o controle do andamento das inscrições em disciplinas feitas por alunos.

Exercício

- ✓ O sistema deve se comunicar com o Sistema de Recursos Humanos para obter dados cadastrais sobre os professores.
- ✓ O sistema deve se comunicar com o Sistema de Faturamento para informar as inscrições realizada pelos alunos.
- ✓ O sistema deve manter informações cadastrais sobre os alunos e sobre seus históricos escolares.